Accelerator Science

Panagiotis Spentzouris
Fermilab
Estelle Cormier
Tech-X

Accelerator Science Drivers

- New techniques and technologies
  - Optimize, evolve concepts, design accelerator facilities based on new concepts
- Maximize performance of “conventional” techniques and technologies
  - Optimize operational parameters, understand dynamics (manipulation and control of beams in full 6D phase space)
- Desirable outcome: achieve higher gradients for energy frontier applications, minimize losses for intensity frontier applications

Computational Accelerator Physics Challenges

- Energy Frontier. Modeling infrastructure to
  - Develop techniques, technologies and materials to achieve higher acceleration gradients
    - dielectric and plasma wave structures, beam cooling
  - Optimize existing technologies
    - Superconducting rf cavities
  - Optimize and test new designs (Muon and Electron Colliders,…)
- Intensity Frontier. Modeling infrastructure to
  - Understand and control instabilities, to minimize and mitigate beam losses
    - Self-fields, wakefields, interaction with materials, geometry and long term tracking accuracy
  - Optimize existing technologies
    - Srf cavities
  - Optimize and test designs (Project-X, …)

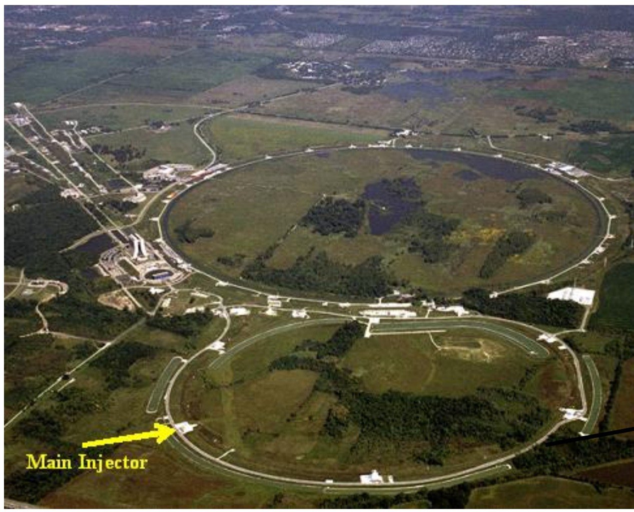
Requirement Investigation

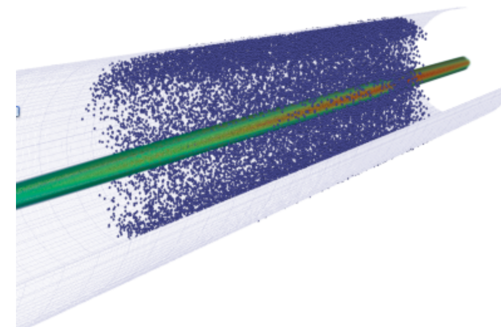
- Energy Frontier *(driver)*
  - LWFA *(10 GeV/m stages)*
    - Multi-scale (laser, plasma, beam), machine design R&D
  - PWFA *(10-100 GeV/m)*
    - Multi-scale (plasma, beam), machine design R&D
  - DLA *(accelerator on a chip, few GeV/m)*
    - Multi-scale, open structures, radiation, structure design R&D, thermal-mechanical-EM
  - Muon colliders *(compact)*
    - Multi-scale (beam, cooling material), machine design R&D, multi-physics
  - Electron colliders *(maximize conventional structure performance, 2-beam acceleration, 100 MeV/m)*
- Intensity Frontier
  - proton Linac *(maximize conventional structure performance)*
    - Wakefields, thermal-mechanical-EM analysis
  - proton Circular
    - Loss mitigation
    - Multi-scale (self-interactions, impedance, clouds…)

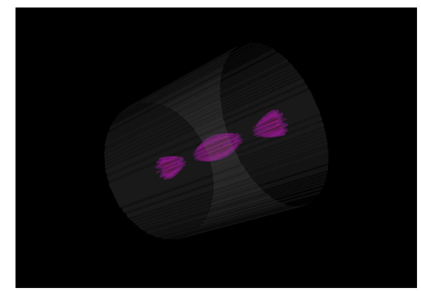
white paper received

no white paper

Example: High-Intensity Proton Drivers

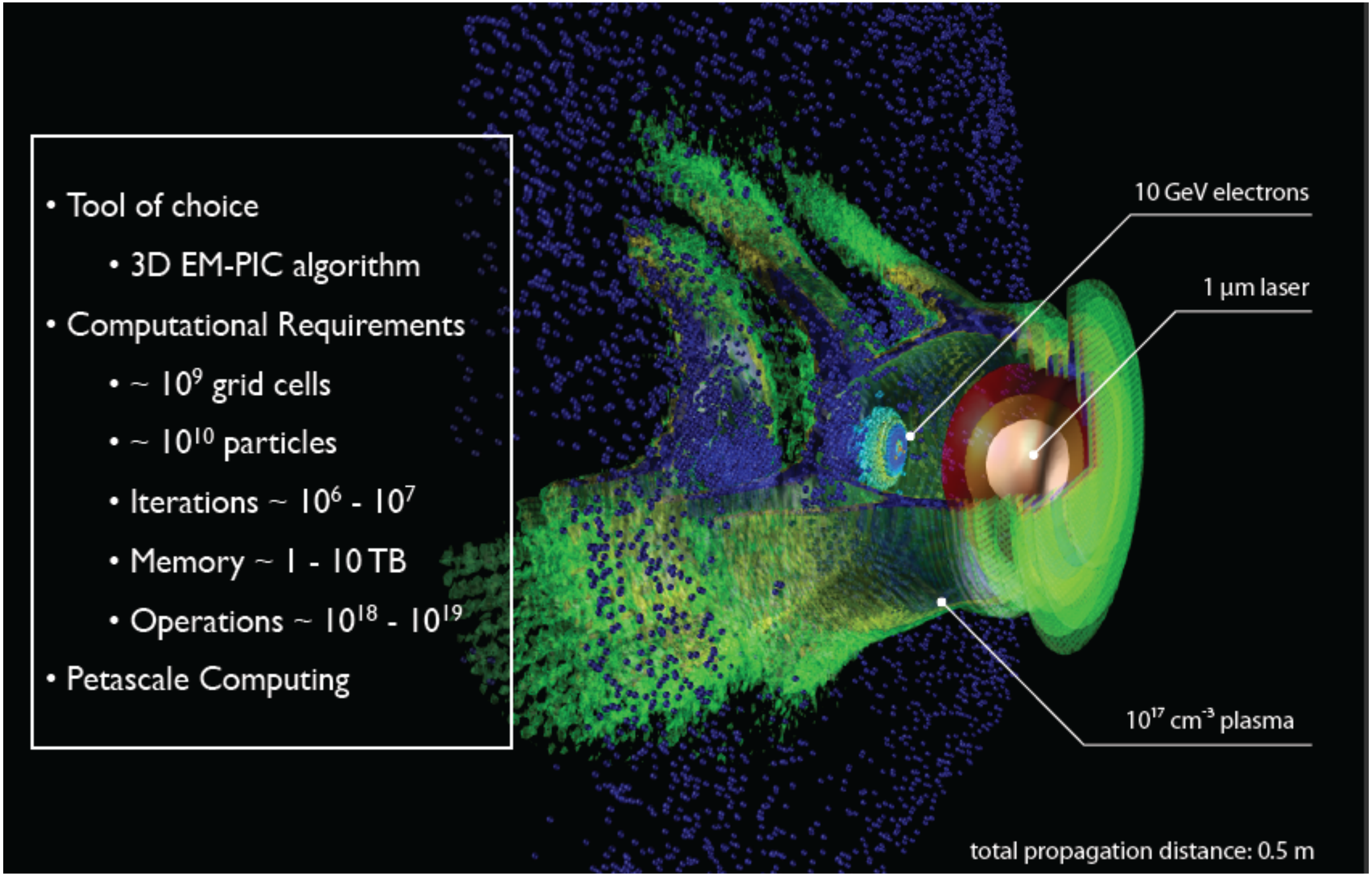






- Wide range of scales:
  - accelerator complex (10^3m) → EM wavelength (10^2-10 m) → component (10-1 m) → particle bunch (10^{-3} m)
  - Need to correctly model intensity dependent effects and the accelerator lattice elements (fields, apertures), to identify and mitigate potential problems due to instabilities that increase beam loss; thousands of elements, millions of revolutions
  - **Calculating 1e-5 losses at 1% requires modeling 1e9 particles, interacting with each-other and the structures around them at every step of the simulation**

Example:LWFA multi-scale physics



Summary of Requirements

- Intensity Frontier accelerator needs
  - beam loss characterization and control
  - control room feedback : need fast turnaround of simulations
  - direct comparison between beam diagnostics detectors and simulated data (need development of new tools)
- Energy Frontier accelerator needs
  - beam stability characterization
  - ability to produce end-to-end designs
  - control room feedback: fast turnaround and fast analysis of massive data (requires development of new, automated, analysis tools)
  - new physics model capabilities (e.g., radiation and scattering)
  - better numerical models (less numerical noise) - identification of most suited numerical techniques
- All Frontiers
  - integrated / multi physics modeling: improve algorithms (integrate more physics in the model), massive computing resources
  - common interfaces, geometry and parameter description, job submission, from tablet to supercomputer.

Summary of Findings

- All frontiers require integrated modeling. Currently, for high-fidelity modeling, because of the many degrees of freedom involved in the problem, we run “single physics” or “few physics” model simulations. This will require better algorithms with the ability to utilize massive computing resource (tightly coupled).
- All frontiers would like “consolidation” of user interfaces and community libraries and tools, including analysis tools
- All frontiers will benefit from coordinated development and computing R&D for a sustainable program

Summary of Findings

- Energy Frontier future accelerators have a lot of specific new physics model capability needs (with some overlap, for example radiation and scattering, which is relevant to muon collider, plasma and gamma-gamma options). Also, Energy Frontier machines require better numerical models (less numerical noise). This has direct impact on the choice of numerical techniques for different physics implementations.

Summary of Findings

- Intensity frontier machines would like "control room feedback" capabilities (because of the loss implications). Would it be possible with utilization of new computing technologies to deliver such fast turnaround? Such capability is also relevant to R&D experimental efforts (for example plasma driven acceleration), but there the requirements are even more stringent because, in addition to faster modeling, extracting useful information out of a simulation run involves analyzing massive data. So this leads to analysis tools development requirements that must perform in almost real time (as the experiment is running).
- For intensity frontier and for experimental programs develop analysis tools and techniques to allow for direct comparison between beam diagnostic detectors (what machine physicists see) and simulated quantities (what computational physicists use for input to the simulation and to describe the output of the models).

Resource and infrastructure needs

INFRASTRUCTURE

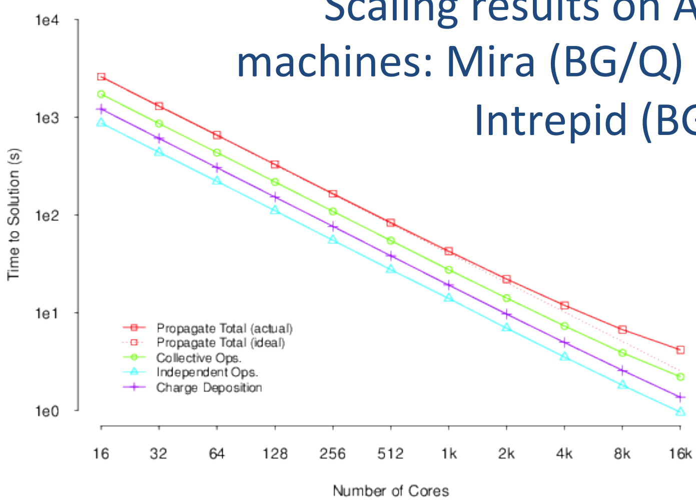
Current methods and tools (partial list)

- Mature set of codes; most already HPC capable, R&D for new architectures. Variety of solvers:
  - <u>Electrostatic:</u> multigrid; AMR multigrid; spectral
  - <u>Electromagnetic:</u> finite element direct and hybrid; extended stencil finite-difference; AMR finite-difference
  - <u>Quasi-static:</u> spectral (*QuickPIC*)
- Specialized codes (beam-material interaction for ionization cooling, …)
- Software Libraries and Tools
  - Chombo, FFTW, Geant4, HDF5, LAPACK, METIS, MPI, MPI/IO, MUMPS, ScaLAPACK, SuperLU, TAU, TAO, Trillinos
  - C++, Fortran90, Python, CUDA
  - Analysis: ParaView, Python, R Language, ROOT, VisIt.
- Although simulated data volume is increasing can't compare with HEP experiment requirements
  - Not a driver in storage, networking, etc. Will leverage HEP experiment motivated advances

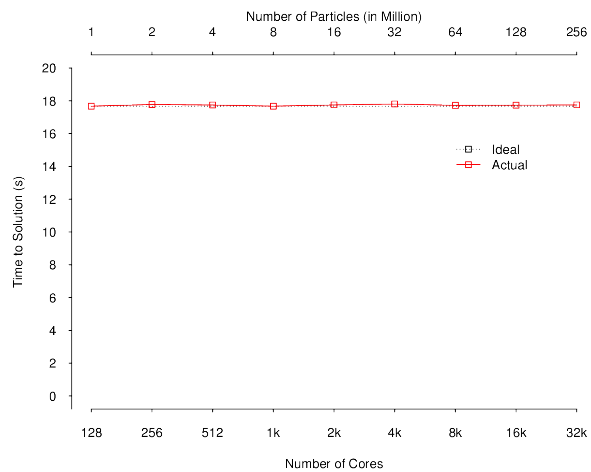
HPC example: scaling on current architectures

- Synergia modeling of proton drivers
  - Single- and multiple-bunch simulations

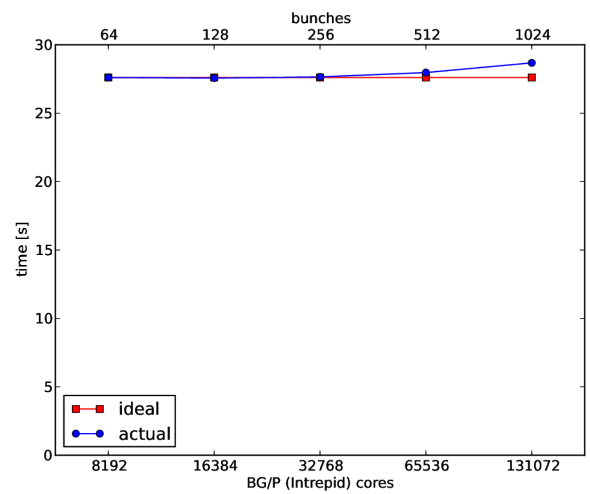
Scaling results on ALCF machines: Mira (BG/Q) and Intrepid (BG/P)



Single-bunch strong scaling from 16 to 16,384 cores
32x32x1024 grid, 105M particles



Weak scaling from 1M to 256M *particles*
128 to 32,768 cores



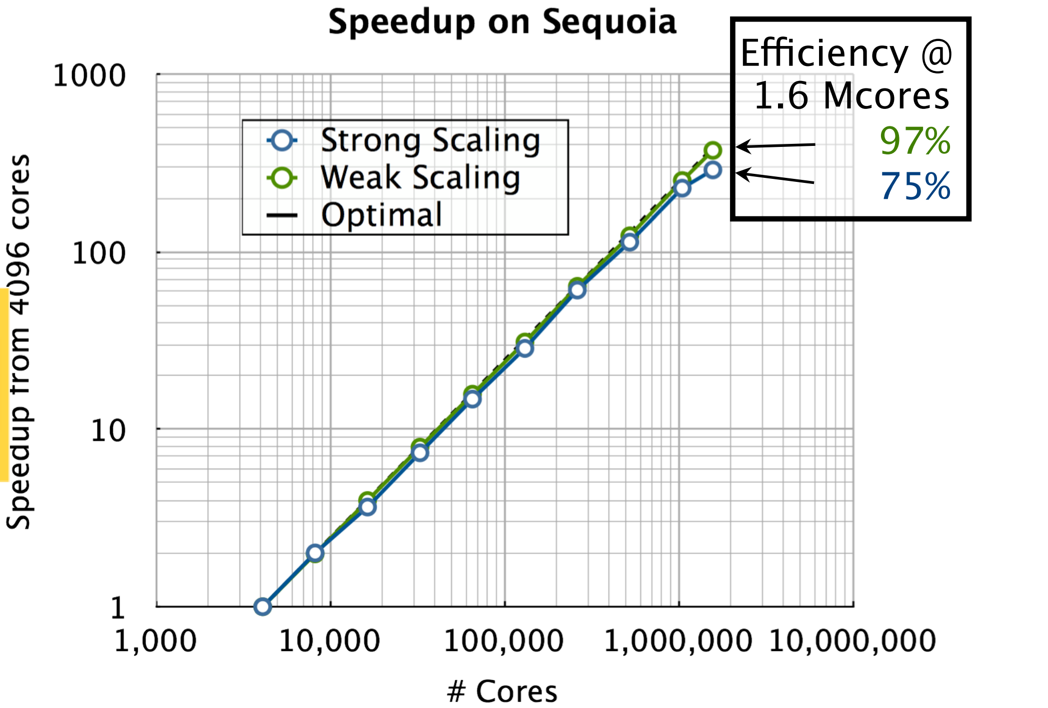
Weak scaling from 64 to 1024 *bunches*
8192 to 131,072 cores
Up to over 10^{10} particles

HPC example: scaling on current architectures

OSIRIS: 1.6 million cores and 2.2 PFLOPS

- Performance tests on Blue Waters
772 480 cores (XE partition)
- Problem size (LWFA)
cells = 38624 × 1024 × 640
400 particles/cell (~ 10^{13})
- Computations
2.2 PFlop/s performance
31% of R_{peak}

UCLA



LLNL Sequoia
IBM BlueGene/Q
#2 – TOP500 Nov/12
1572864 cores
R_{max} 16.3 PFlop/s

Emerging technology research

- GPUs and Multicore
  - Shared memory is back
  - Some things could get easier, some are harder
    - Charge deposition in shared memory systems is the key challenge (PIC example)
- Multi-level parallelism very compatible with current communication avoidance approaches, but change in programing paradigm needed
- Could provide solutions for integrated high-fidelity modeling
  - Multi-physics and parameter optimization
- Continuing R&D necessary

Example of current R&D: GPU-accelerated results

- Benchmark with 2048x2048 grid, 150,994,944 particles, 36 particles/cell
- optimal block size = 128, optimal tile size = 16x16. Single precision
- GPU algorithm also implemented in OpenMP

- Electrostatic
- mx=16, my=16, dt=0.1
- Total speedup was about 35 compared to 1 CPU, and about 3 compared to 12 CPUs.

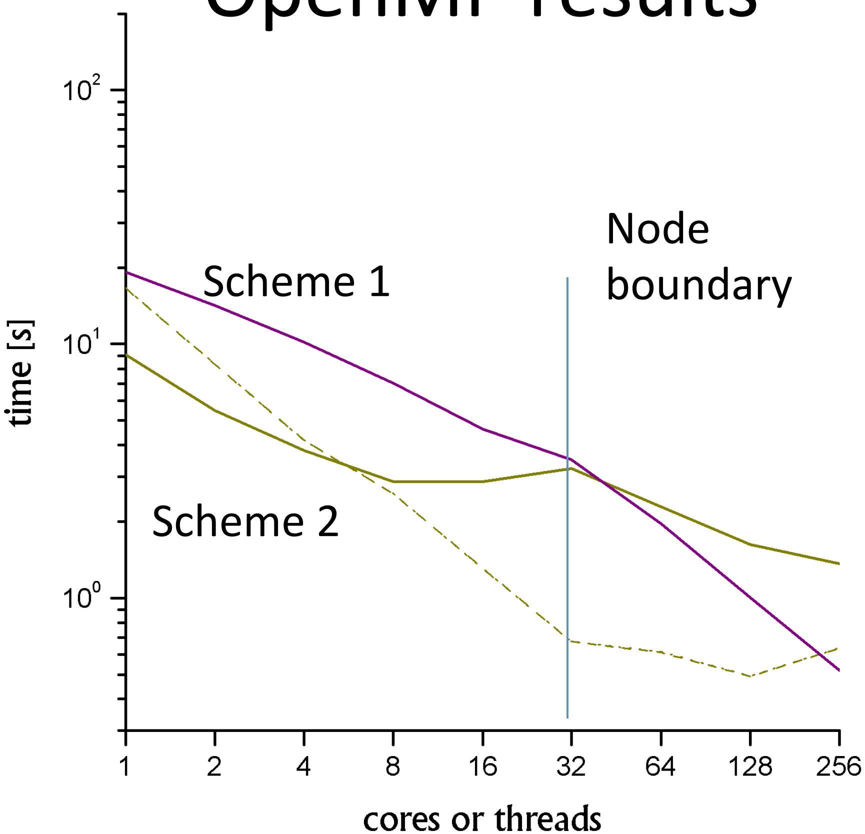
	CPU:Inteli7	GPU:M2090	OpenMP(12 cores)
Push	22.1 ns	0.532 ns	1.678 ns
Deposit	8.5 ns	0.227 ns	0.818 ns
Reorder	0.4 ns	0.115 ns	0.113 ns
Total Particle	31.0 ns	0.874 ns	2.608 ns

- Electromagnetic
- mx=16, my=16, dt=0.04, c/vth=10
- Total speedup was about 51 compared to 1 CPU, and about 4 compared to 12 CPUs.

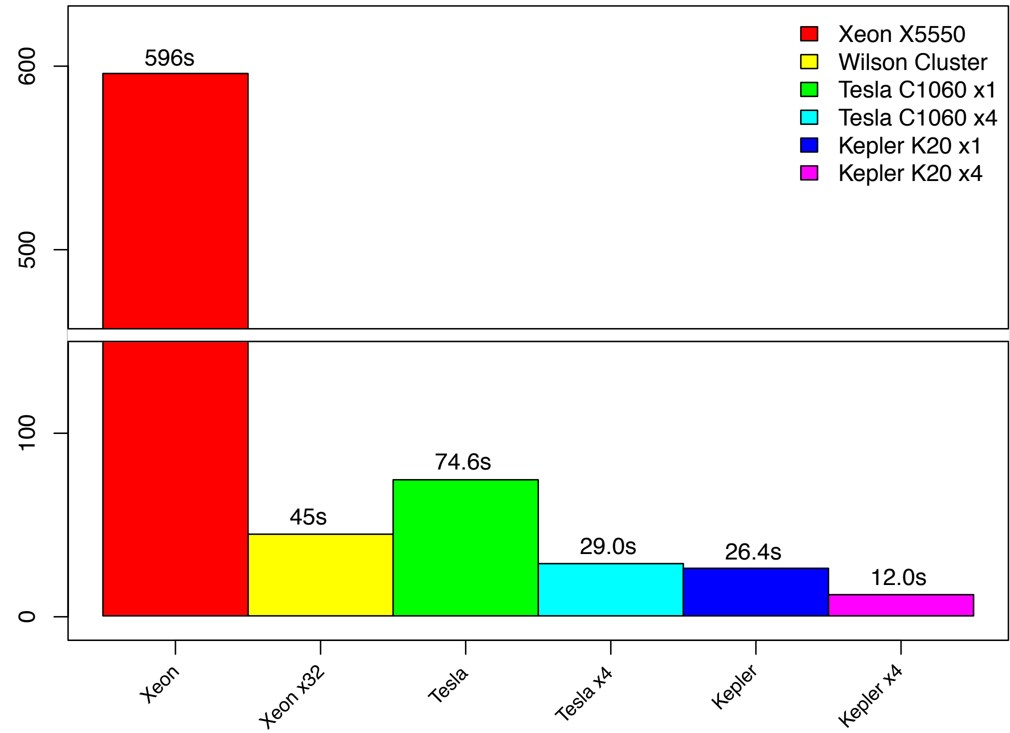
	CPU:Inteli7	GPU:M2090	OpenMP (12 cores)
Push	66.5 ns	0.426 ns	5.645 ns
Deposit	36.7 ns	0.918 ns	3.362 ns
Reorder	0.4 ns	0.698 ns	0.056 ns
Total Particle	103.6 ns	2.042 ns	9.062 ns

Example of current R&D: GPU and multicore results

OpenMP results



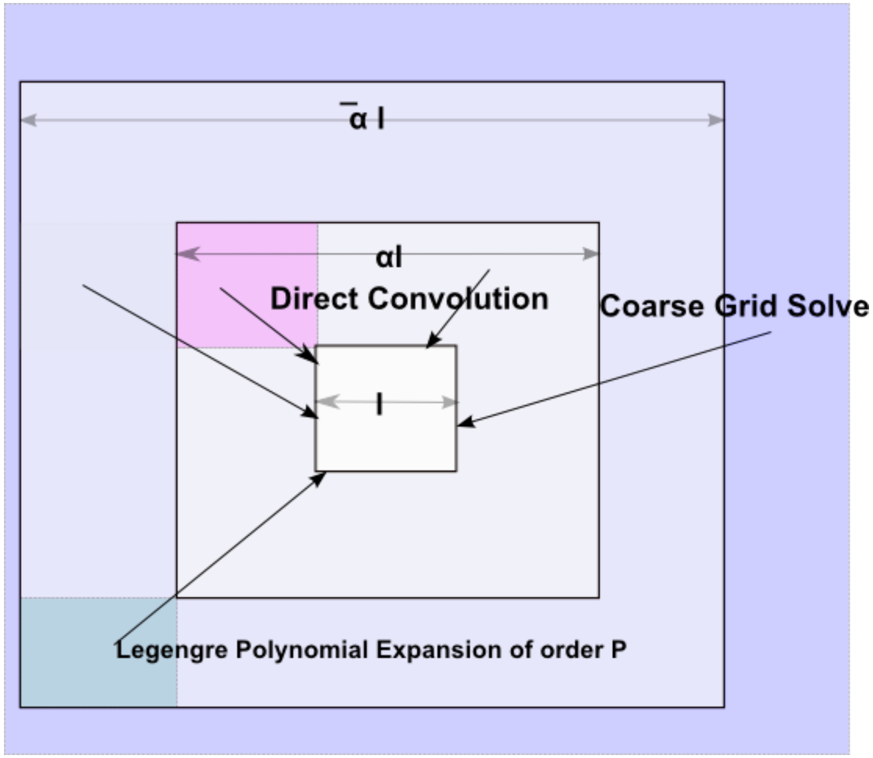
GPU results



Fermilab

Advanced algorithms: method of local corrections

- Potential-theoretic domain - decomposition Poisson solver compatible with AMR grids
- One V-cycle solver
  - Downsweep: build RHS for coarser grids using discrete convolutions and Legendre polynomial expansions
    - exploits higher-order FD property of localization
    - Convolutions performed with small FFTs and Hockney 1970
  - Coarse solve
    - Either MLC again, or FFT
  - Upsweep
    - Solve for Φ_h on boundary of patch
    - Interpolation and summations
    - Local Discrete Sine Transform Solve



No iteration, accurate, no self-force problems, large number of flops per unit of communication (messages and DRAM).

BERKELEY LAB
LAWRENCE BERKELEY NATIONAL LABORATORY

New Architectures Needs

- Opportunity for multi-scale, multi-physics modeling and "near-real-time" feedback, if could be used efficiently
- Work on algorithmic and code development to enable our codes to perform on new architectures ("lightweight" processors, accelerators, or hybrid configurations).
- Current strategy is to abstract and parameterize data structures so that are portable and enable efficient flow of data to a large number of processing units in order to maintain performance.
  - Already ported subset of solvers and PIC infrastructure on the GPU
  - Evaluate current approach, develop workflow tools and frameworks
  - Investigate new algorithms and approaches
- As technology evolves, we need specs of new production systems and availability of test machines well in advance of deployment
- Effort should be funded and coordinated as a program, to develop common set of libraries and interfaces and avoid duplication

Summary

- Multi-physics modeling necessary to advance accelerator science. Requires
  - Frameworks to support advanced workflows
  - Efficient utilization of large computing resources, HPC in many cases
- Evolving technologies (light-weight CPU plus accelerators) require R&D and could result in significant changes
  - Advanced algorithmic research underway, will require continuing support
  - Programmatic coordination necessary to efficiently utilize resources
- Not a driver in data and networking, although could benefit from utilization of collaborative tools
  - Will benefit from advances driven by HEP experiment needs